Enhancing the Differential Evolution with Convergence Speed Controller for Continuous Optimization Problems

Shujin Ye
South China University of Technology
yeshujin2@163.com

Han Huang*
South China University of Technology
hhan@scut.edu.cn

Changjian Xu
South China University of Technology
x.changjian@mail.scut.edu.cn

ABSTRACT

In this paper, we proposed a convergence speed controller (denoted as CSC) framework to improve the performance of differential evolution for continuous optimization problems.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*

Keywords

convergence speed controller; differential evolution; locally optimal

1. SUMMARY

In this paper, we introduced a DE algorithm with convergence speed controller (CSC) framework for continuous optimization problem, which can generally avoid premature convergence. According to the cosine similarity and the relative fitness error between two randomly selected individuals from the DE population, CSC provides a period strategy to detect whether the DE stagnates local optimums. Once the cosine similarity and fitness values are out of the threshold bound, CSC will regenerate the population on the basis of N top ranking individuals (denoted as *topN*) to protect the DE from premature convergence. Moreover, CSC implements a mechanism for unimproved individuals, which will regenerate the individual on the basis of the best-so-far individual, to accelerate the convergence velocity of the target individual. The test results of 25 standard numerical benchmark functions reveal that the proposed algorithm outperforms three other DE algorithms.

2. THE CSC-DE ALGORITHM

The pseudo code of CSC-DE is illustrated by Table 1.

*This author is the corresponding author.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
ACM 978-1-4503-2881-4/14/07.
http://dx.doi.org/10.1145/2598394.2598501.

Table 1: the pseudo code of CSC-DE

```
{x_i ... i-th individual of population}
{MaxIte ... maximum number of Iterations }
{x_best ... the best-so-far solution}
Initialization()
for ite = 0; ite < MaxIte; ite = ite + 1 do
    {**Update the topN**}
    {**Perform one iteration of the DE**}
    if condition1 is met then
        Run Rule1: renew x_1, x_2, ..., x_NP to slow down
        the convergence speed;
    end if
    if condition2 is met then
        Run Rule2: renew the stopping individual x_i to
        accelerate the convergence speed;
    end if
    Update x_best
end for
```

2.1 Initialization

CSC-DE starts with a population of NP D-dimensional individuals within the search space constrained by the prescribed minimum and maximum bounds $\vec{X}_{min} = [x_{1,min}, x_{2,min}, \cdots, x_{D,min}]$ and $\vec{X}_{max} = [x_{1,max}, x_{2,max}, \cdots, x_{D,max}]$. The initial value of the j-th component of the i-th individual at generation G=0 as

$$x_{i,j,0} = x_{j,min} + rand(0,1) \cdot (x_{j,max} - x_{j,min}) \quad (1)$$

where $rand(0,1)$ is a uniformly distributed random variable within the range $[0,1]$.

2.2 History information based $topN$ updating

In order to record the information during the optimization process, we designed a set named $topN$ which contains N excellent individuals. Moreover, to make all the individuals in $topN$ cover a wide search space, the cosine similarity between any two individuals in $topN$ is set to less than sim(sim is a threshold). Given two individuals x_a and x_b, their cosine similarity is calculated by Expression (2).

$$cos(x_a, x_b) = \frac{\sum_{j=1}^{D} x_{a,j} \cdot x_{b,j}}{\sqrt{\sum_{j=1}^{D} x_{a,j}^2} \cdot \sqrt{\sum_{j=1}^{D} x_{b,j}^2}} \quad (2)$$

Once CSC-DE detects out that DE falls into a local minima possibly, it can generate a new population according to the information of the individuals in $topN$.

2.3 Condition 1 and Rule 1

One of the most accepted idea to detect convergence is that individuals of the entire population are very similar in position and fitness to each other. Therefore, we use cosine similarity and the ratio of two randomly selected individuals for the convergence condition. Hence, the condition of convergence detection (condition 1 in Table 1) is designed as Expression (3).The condition of convergence detection (condition 1 in Table 1) is designed as Expression (3).

$$\begin{cases} cos(x_a, x_b) > sim \\ \frac{|f(x_a)-f(x_b)|}{f(x_a)} < dif \end{cases} \tag{3}$$

where sim and dif are threshold values. Index $a \in \{1, \ldots, NP\}$ and $b \in \{1, \ldots, NP\}$ are randomly chosen integers($a \neq b$).

2.4 Condition 2 and Rule 2

Rapid convergence to better solution is regarded as an advantage of DE. Besides Rule 1 plays a role as convergence speed slower, an acceleration of convergence speed is also necessary to make CSC more heuristic and robust. For the acceleration, Rule 2 will be implemented when condition 2 is satisfied. As can be seen from the process of DE algorithm, it may happen that an individual unchanged after a number of iterations. This may slow down the convergence of the algorithm. Therefore, condition 2 is presented as

$$t_i > \tau \tag{4}$$

where the range of index i is $\{0, \ldots, NP\}$. t_i denotes the number of consecutive iterations that the ith individual has not been improved. And τ is a threshold value.

3. EXPERIMENTS AND RESULTS

3.1 Algorithms Compared and Parametric Set-up

CSC-DE algorithm is tested with twenty-five test problems from CEC-2005 benchmark functions[1]. The performance of the CSC-DE algorithm is compared with the DE/current-to-best/1/bin[2], JADE [3] and MDE_ρBX [4]. To make the comparison fair, we adopt the best parameter settings from their respective literature.

3.2 Results on Numerical Benchmarks

Tables 5 show the mean and the standard deviation of the best-of-run errors for 25 independent runs of each of the four algorithms on 25 benchmark benchmarks. As can be seen, CSC-DE defeated DE/current-to-best/1/bin in almost all functions but f_{24} and f_{25}. In short, these simulation results show that CSC-DE can greatly improve the original algorithm's global convergence ability and running accuracy. Compared to all other algorithms, CSC-DE managed to win on 17 functions. It is competitive compared to state-of-the-art algorithms

3.3 Acknowledgments

This work is supported by National Natural Science Foundation of China (61370102,61170193, 61203310), Guangdong Natural Science Foundation (S2012010010613), the Fundamental Research Funds for the Central Universities, SCUT(2014ZG0043)and The Pearl River Science & technology Star Project (2012J2200007).

Table 2: MEAN AND STANDARD DEVIATION OF VALUES OBTAINED FROM DE/current-to-best/i/bin, JADE, MDE_ρBX AND CSC-DE ON 2005 BENCHMARKS WITH D=30.

	DE/current-to-best/i/bin	JADE	MDE_ρBX	CSC-DE
f01	0.00E+00 (0.00E+00)	**0.00E+00** **(0.00E+00)**	0.00E+00 (0.00E+00)	**0.00E+00** **(0.00E+00)**
f02	2.21E-07 (1.858E-07)	1.02E-26 (8.42E-27)	2.40E-08 (8.21E-08)	**1.79E-27** **(5.10E-28)**
f03	2.55E+05 (2.05E+05)	**1.03E+04** **(7.07E+03)**	1.20E+06 (384621.9)	5.37E+04 (39797.17)
f04	0.010474 (0.011039)	1.72E-09 (5.60E-09)	1.16E+00 (1.65E+00)	**1.26E-11** **(5.21E-11)**
f05	3.76E+01 (4.42E+01)	**1.19E-07** **(4.17E-07)**	2.86E+03 (4.72E+02)	3.03E+01 (3.00E+01)
f06	9.57E-01 (1.73E+00)	9.35E+00 (2.55E+01)	5.43E+04 (1.50E+05)	**2.05E-06** **(7.08E-06)**
f07	5.91E-03 (0.007621)	5.63E+03 (5.60E+03)	1.06E+01 (1.98E+01)	**1.18E-03** **(3.34E-03)**
f08	2.10E+01 (4.32E-02)	2.08E+01 (1.93E-01)	2.09E+01 (4.76E-02)	**2.00E+01** **(1.54E-02)**
f09	1.14E+02 (6.11E+01)	**0.00E+00** **(0.00E+00)**	1.28E+01 (3.63E+00)	1.14E+01 (4.33E+00)
f10	1.92E+02 (1.73E+01)	**2.56E+01** **(4.32E+00)**	2.94E+01 (14.1601)	4.23E+01 (12.526)
f11	3.94E+01 (1.82E+00)	2.51E+01 (1.54E+00)	2.58E+01 (6.08E+00)	**4.24E+00** **(1.48E+00)**
f12	9.69E+05 (1.03E+05)	6.51E+03 (4.76E+03)	1.14E+04 (8.77E+03)	**5.86E+03** **(6.95E+03)**
f13	1.56E+01 (1.13E+00)	**1.44E+00** **(1.47E-01)**	2.98E+00 (6.75E-01)	2.83E+00 (4.56E-01)
f14	1.39E+01 (1.37E-01)	1.21E+01 (4.27E-01)	1.21E+01 (1.02E+00)	**1.18E+01** **(2.74E-01)**
f15	3.64E+02 7.95E+01	3.21E+02 (9.70E+01)	3.20E+02 (5.79E+01)	**2.39E+02** **(2.72E+01)**
f16	2.40E+02 1.01E+02	1.16E+02 (1.40E+02)	7.74E+01 (9.99E+01)	**5.18E+01** **(7.91E+00)**
f17	2.80E+02 9.34E+01	1.87E+02 (1.03E+02)	1.71E+02 (2.03E+02)	**1.67E+02** **(1.51E+02)**
f18	8.97E+02 2.88E+01	9.04E+02 (1.37E+00)	9.15E+02 (5.17E+00)	**8.77E+02** **(4.90E+01)**
f19	8.95E+02 3.20E+01	9.04E+02 (1.80E-01)	9.14E+02 (3.68E+00)	**8.00E+02** **(3.58E-02)**
f20	8.93E+02 3.47E+01	9.04E+02 (6.10E-01)	9.11E+02 (2.33E+01)	**8.64E+02** **(5.33E+01)**
f21	5.00E+2 (00E+00)	5.00E+02 (1.74E-01)	5.01E+02 (4.81E+00)	**5.00E+02** **(8.20E-03)**
f22	9.21E+02 9.49E+00	**8.56E+02** **(1.49E+01)**	9.02E+02 (1.33E+01)	8.74E+02 (3.70E+01)
f23	6.16E+02 1.63E+02	5.34E+02 (1.60E-01)	5.34E+02 (3.46E-01)	**5.34E+02** **(1.27E-03)**
f24	**2.00E+02** **(00E+00)**	2.00E+02 (1.53E+02)	2.00E+02 (9.17E-05)	2.00E+02 (9.10E-03)
f25	2.10E+02 5.90E-01	1.27E+03 (1.20E+01)	**2.01E+02** **(3.61E+00)**	2.11E+02 (9.94E-01)

4. REFERENCES

[1] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization, KanGAL Report 2005005.

[2] E. Mezura-Montes, J. Velázquez-Reyes, C. Coello Coello, Modified differential evolution for constrained optimization, in: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, IEEE, 2006, pp. 25–32.

[3] J. Zhang, A. C. Sanderson, Jade: adaptive differential evolution with optional external archive, Evolutionary Computation, IEEE Transactions on 13 (5) (2009) 945–958.

[4] S. M. Islam, S. Das, S. Ghosh, S. Roy, P. N. Suganthan, An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42 (2) (2012) 482–500.